

# C S - COMPUTER SCIENCE

## Computer Science: C S

### Lower-Division Courses

#### C S 301K. Foundations of Logical Thought.

Introductory logic in the context of computing; introduction to formal notations; basic proof techniques; sets, relations, and functions. Three lecture hours a week for one semester. Some sections also require one discussion hour a week.

#### C S 302. Computer Fluency.

An introduction to the fundamental concepts of computing: how computers work, what they can do, and how they can be used effectively. Some programming is required. Three lecture hours or two lecture hours and one discussion hour a week for one semester. Credit for [Computer Science 302](#) may not be earned after a student has received credit for [Computer Science 303E](#), [312](#), [312H](#), [314](#), or [314H](#). May not be counted toward a degree in computer science.

#### C S 303E. Elements of Computers and Programming.

Problem solving and fundamental algorithms for various applications in science, business, and on the World Wide Web, and introductory programming in a modern object-oriented programming language. The equivalent of three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 303E](#), [312](#), [312H](#). Credit for [Computer Science 303E](#) may not be earned after a student has received credit for [Computer Science 314](#), or [314H](#). May not be counted toward a degree in computer science.

#### C S 103F. Ethical Foundations of Computer Science.

An introduction to reasoning about the effects of the technology created by computer scientists and understanding the ethical and real-world consequences of that technology. Emphasizes direct participation and discussion using case studies to illustrate ethical topics. One lecture hour a week for one semester. [Computer Science 103F](#) and [109](#) (Topic: Ethical Foundations of Computer Science) may not both be counted.

#### C S 104C. Competitive Programming.

Explores the algorithms needed to solve a given problem, reason about corner cases, and write correct implementations under time pressure. Focuses on skills useful for international programming competitions, whiteboard job interviews, and programming tasks that require quickly solving challenging problems with creative use of algorithms. One lecture hour a week for one semester. [Computer Science 104](#) and [109](#) (Topic: Competitive Programming) may not both be counted. Prerequisite: [Computer Science 311](#) or [311H](#) with a grade of at least C-, and registration or credit in [Computer Science 314](#) or [314H](#) with a grade of at least C-.

#### C S 105C. Computer Programming: C++.

Explores the syntax and concepts needed to program in C++. One lecture hour a week for one semester. Prerequisite: [Computer Science 313E](#), [314](#), [314H](#), or Electrical Engineering [422C](#), with a grade of at least C-.

#### C S 105P. Topics In Computer Programming Languages.

An introduction to programming in a particular computer language. Includes designing and implementing programs. One lecture hour a week for one semester. May be repeated for credit when the topics vary. Prerequisite: [Computer Science 313E](#), [314](#), [314H](#), or Electrical Engineering [422C](#), with a grade of at least C-, and consent of instructor.

#### C S 108. Software Systems.

Introduction to the use of a particular software system. Students build applications that exploit the system being studied. One lecture hour a week for one semester. May be repeated for credit when the topics vary. Offered on the pass/fail basis only. Prerequisite: [Computer Science 313E](#), [314](#), [314H](#), or Electrical Engineering [422C](#), with a grade of at least C-.

#### C S 109, 209, 309. Topics in Computer Science.

For each semester hour of credit earned, one lecture hour a week for one semester. May be repeated for credit when the topics vary. Prerequisite: Varies with the topic.

**Topic 1: Quantum Computing I.** Introduction to quantum information and communication. Computer Science 309 (Topic: Quantum Computing-FRI) and 309 (Topic 1) may not both be counted.

#### C S 311. Discrete Mathematics for Computer Science.

Restricted to computer science majors. A focus on discrete mathematical tools of fundamental importance to the working computer scientist. An emphasis is placed on using logical notation to express rigorous mathematical arguments. Subjects include proof by induction, introduction to graph theory, recurrences, sets, functions, and an introduction to program correctness. Three lecture hours a week for one semester. [Computer Science 311](#) and [311H](#) may not both be counted. Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 312](#) or [312H](#); and [Mathematics 408C](#), [408K](#), or [408N](#), or registration for [Mathematics 408C](#).

#### C S 311H. Discrete Mathematics for Computer Science: Honors.

Restricted to honors computer science majors. A focus on discrete mathematical tools of fundamental importance to the working computer scientist. An emphasis is placed on using logical notation to express rigorous mathematical arguments. Subjects include proof by induction, graph theory, number theory, combinatorics, recurrences, sets, functions, an introduction to program correctness, and discrete probability. Three lecture hours a week for one semester. [Computer Science 311](#) and [311H](#) may not both be counted. Prerequisite: [Mathematics 408K](#) or [408N](#) with a grade of at least C-; or registration or credit with at least C- in [Mathematics 408C](#).

#### C S 312. Introduction to Programming.

First part of a two-part sequence in programming. Fundamental concepts of structured programming; procedures and data structures with a focus on problem solving strategies and implementation; introduction to concepts of informal specification, informal reasoning about program behavior, debugging, and ad hoc testing. Three lecture hours and one discussion hour a week for one semester. Only one of the following courses may be counted: [Computer Science 303E](#), [312](#), [312H](#). Credit for [Computer Science 312](#) may not be earned after a student has received credit for [Computer Science 314](#) or [314H](#). Prerequisite: Credit with a grade of at least C- or registration for [Mathematics 408C](#), [408K](#), or [408N](#).

### **C S 312H. Introduction to Programming: Honors.**

Restricted to computer science majors. First part of a two-part sequence in programming. Fundamental concepts of structured programming; procedures and data structures with a focus on problem solving strategies and implementation; introduction to concepts of informal specification, informal reasoning about program behavior, debugging, and ad hoc testing. Three lecture hours and one discussion hour a week for one semester. Only one of the following courses may be counted: [Computer Science 303E](#), [312](#), [312H](#). Credit for [Computer Science 312H](#) may not be earned after a student has received credit for [Computer Science 314](#) or [314H](#). Prerequisite: Credit with a grade of at least C- or registration for [Mathematics 305G](#), or a passing score on the College of Natural Sciences mathematics placement examination; and consent of the honors director.

### **C S 313E. Elements of Software Design.**

Object-oriented design of software using Python, a modern high-level language. Introduction to abstract data types, fundamental algorithms, and basics of complexity analysis. Three lecture hours a week for one semester. May not be counted toward a degree in computer science. Prerequisite: [Computer Science 303E](#), [312](#), or [312H](#) with a grade of at least C-.

### **C S 314. Data Structures.**

Second part of a two-part sequence in programming. Introduction to specifications, simple unit testing, and debugging; building and using canonical data structures; algorithm analysis and reasoning techniques such as assertions and invariants. Three lecture hours and one discussion hour a week for one semester. [Computer Science 314](#) and [314H](#) may not both be counted. Prerequisite: [Computer Science 312](#) or [312H](#) with a grade of at least C-.

### **C S 314H. Data Structures: Honors.**

Second part of a two-part sequence in programming. Introduction to specifications, simple unit testing, and debugging; building and using canonical data structures; algorithm analysis and reasoning techniques such as assertions and invariants. Three lecture hours and one discussion hour a week for one semester. [Computer Science 314](#) and [314H](#) may not both be counted. Prerequisite: Consent of honors faculty advisor.

### **C S 118H. Introduction to Computer Science Research: Honors.**

Restricted to computer science honors students. Introduction to computer science research. Discuss various aspects of undergraduate research, including the benefits of undergraduate research, ways to get started in undergraduate research, the mechanics and expectations of an undergraduate honors thesis, and tips to help have a successful undergraduate research experience. One lecture hour a week for one semester. [Computer Science 118H](#) and [178H](#) (Topic: Intro to Comp Sci Rsch: Honors) may not both be counted. Prerequisite: Consent of instructor.

### **C S 119S, 219S, 319S, 419S, 519S, 619S, 719S, 819S, 919S. Topics in Computer Science.**

This course is used to record credit the student earns while enrolled at another institution in a program administered by the University's Study Abroad Office. Credit is recorded as assigned by the study abroad adviser in the Department of Computer Science. University credit is awarded for work in an exchange program; it may be counted as coursework taken in residence. Transfer credit is awarded for work in an affiliated studies program. May be repeated for credit when the topics vary.

## **Upper-Division Courses**

### **C S 120N, 220N, 320N. Topics in Computer Science for Nonmajors.**

For each semester hour of credit earned, one lecture hour a week for one semester. May not be counted toward a degree in computer science. May be repeated for credit when the topics vary. Prerequisite: Varies with the topic.

### **C S 323E. Elements of Scientific Computing.**

Explore the fundamentals of software issues related to scientific computing. Discuss subjects including floating-point computations, numerical computation errors, interpolation, integration, solution of linear systems of equations, optimization, and initial value problems of ordinary differential equations. Examine implementation of algorithms that are investigated using Python for all scientific computations. Three lecture hours a week for one semester. [Computer Science 323E](#) and [323H](#) may not both be counted. Prerequisite: [Computer Science 313E](#), [314](#), or [314H](#) with a grade of at least C-.

### **C S 323H. Elements of Scientific Computing: Honors.**

Fundamentals of software issues related to scientific computing. Subjects include floating-point computations, numerical computation errors, interpolation, integration, solution of linear systems of equations, optimization, and initial value problems of ordinary differential equations. Implementation of algorithms are investigated using MATLAB for matrix and vector computations. Examples are drawn from a variety of science and mathematics areas. Three lecture hours a week for one semester. May not be counted toward a degree in computer science. [Computer Science 323E](#) and [323H](#) may not both be counted. Prerequisite: [Computer Science 303E](#), [312](#), or [312H](#) with a grade of at least C-; credit with a grade of at least C- or registration for [Mathematics 340L](#), or [341](#); and consent of the honors director.

### **C S 324E. Elements of Graphics and Visualization.**

Basics of two- and three-dimensional computer graphics systems, modeling and rendering, and selected graphics software APIs. Other topics may include interactive graphics, animation, graphical user interfaces, and the graphical presentation of information. Three lecture hours a week for one semester. May not be counted toward a degree in computer science. Prerequisite: [Computer Science 313E](#), [314](#), or [314H](#) with a grade of at least C-.

### **C S 326E. Elements of Networking.**

Introduction to the principles and basic concepts of the Internet, networking applications and protocols, and simple client/server applications. Other topics may include network technologies and topologies, packet and circuit switching, LANS and WANS, Internet

security, and network management. Three lecture hours a week for one semester. May not be counted toward a degree in computer science. Prerequisite: [Computer Science 313E](#), [314](#), or [314H](#) with a grade of at least C-.

### **C S 327E. Elements of Databases.**

A practical introduction to database management systems, with discussion of database administration and management. Survey of logical modeling, database design with a focus on relational databases, SQL query language, and current applications. Subjects may include data integrity, performance, concurrency, transaction processing, recovery, security, and Web applications. Three lecture hours a week for one semester. May not be counted toward a degree in computer science. Prerequisite: [Computer Science 313E](#), [314](#), or [314H](#) with a grade of at least C-.

### **C S 328E. Topics in Elements of Computing.**

Three lecture hours a week for one semester. May not be counted toward a degree in computer science. May be repeated for credit when the topics vary. Prerequisite: [Computer Science 303E](#), [312](#), or [312H](#) with a grade of at least C-.

### **C S 429. Computer Organization and Architecture.**

Restricted to computer science majors. An introduction to low-level computer design ranging from the basics of digital design to the hardware/software interface for application programs. Includes basic systems principles of pipelining and caching, and requires writing and understanding programs at multiple levels. Four lecture hours and two discussion hours a week for one semester. [Computer Science 429](#) and [429H](#) may not both be counted. Prerequisite: The following courses with a grade of at least C-: [Computer Science 311](#) or [311H](#); and [Computer Science 314](#) or [314H](#).

### **C S 329E. Advanced Topics in Elements of Computing.**

Three lecture hours a week for one semester. May not be counted toward a degree in computer science. May be repeated for credit when the topics vary. Prerequisite: [Computer Science 313E](#), [314](#), or [314H](#) with a grade of at least C-.

### **C S 429H. Computer Organization and Architecture: Honors.**

Restricted to computer science majors. An introduction to low-level computer design ranging from the basics of digital design to the hardware/software interface for application programs. Includes basic systems principles of pipelining and caching, and requires writing and understanding programs at multiple levels. Four lecture hours and two discussion hours a week for one semester. Only one of the following may be counted: [Computer Science 310](#), [310H](#), [429](#), [429H](#). Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 311](#) or [311H](#); [Computer Science 314](#) or [314H](#).

### **C S 129S, 229S, 329S, 429S, 529S, 629S, 729S, 829S, 929S. Topics in Computer Science.**

This course is used to record credit the student earns while enrolled at another institution in a program administered by the University's Study Abroad Office. Credit is recorded as assigned by the study abroad adviser in the Department of Computer Science. University credit is awarded for work in an exchange program; it may be counted as coursework taken in residence. Transfer credit is awarded for work in an affiliated studies program. May be repeated for credit when the topics vary.

### **C S 329W. Cooperative Computer Science.**

This course covers the work period of computer science students in the Cooperative Education program, which provides supervised work experience by arrangement with the employer and the supervising instructor. Forty laboratory hours a week for one semester. The student must repeat the course each work period and must take it twice to receive credit toward the degree; at least one of these registrations must be during a fall or spring semester. However, no more than three semester hours may be counted toward the major requirement. The student's first registration must be on the pass/fail basis; the second must be on the letter-grade basis. Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 311](#) or [311H](#); [314](#) or [314H](#); [429](#), or [429H](#); and consent of the undergraduate adviser.

### **C S 330E. Elements of Software Engineering I.**

Explore software engineering using various programming languages. Use tools to improve the quality of software development. Three lecture hours a week for one semester. [Computer Science 329E](#) (Topic: Elms of Software Engineering) and [330E](#) may not both be counted. Prerequisite: [Computer Science 313E](#), [314](#), or [314H](#) with a grade of at least C-.

### **C S 330F. F1/10 Autonomous Driving.**

Introduction to algorithms required for autonomous driving, and more generally, autonomous mobile robots. Discuss motion control (model-predictive control, time-optimal control), sensing (filtering, clustering), planning (local obstacle avoidance, heuristic search, RRTs), and state estimation (particle filters). Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 330F](#), [330H](#), [378](#) (Topic: F1/10 Autonomous Driving), [378H](#) (Topic: F1/10 Autonomous Driving: Hon). Prerequisite: For computer science majors, [Computer Science 429](#) or [429H](#) with a grade of at least C-; for others, consent of instructor.

### **C S 330H. F1/10 Autonomous Driving: Honors.**

Introduction to algorithms required for autonomous driving, and more generally, autonomous mobile robots. Discuss motion control (model-predictive control, time-optimal control), sensing (filtering, clustering), planning (local obstacle avoidance, heuristic search, RRTs), and state estimation (particle filters). Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 330F](#), [330H](#), [378](#) (Topic: F1/10 Autonomous Driving), [378H](#) (Topic: F1/10 Autonomous Driving: Hon). Prerequisite: For computer science majors, [Computer Science 429](#) or [429H](#) with a grade of at least C-; for others, consent of instructor.

### **C S 331. Algorithms and Complexity.**

An investigation of algorithmic paradigms: divide and conquer, dynamic programming, greedy algorithms, graph algorithms, randomized algorithms, undecidability, NP-completeness, and approximation algorithms. Three lecture hours and one discussion hour a week for one semester. Only one of the following may be counted: [Computer Science 331](#), [331H](#), [378](#) (Topic: Algorithms and Complexity). Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [Mathematics 362K](#) or [Statistics and Data Sciences 321](#); and credit with a grade of at least C- or registration for: [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 331E. Elements of Software Engineering II.**

Explore advanced aspects of software engineering using various programming languages. Use tools to improve the quality of software development. Three lecture hours a week for one semester. Prerequisite: [Computer Science 330E](#) with a grade of at least C-.

### **C S 331H. Algorithms and Complexity: Honors.**

An investigation of algorithmic paradigms: divide and conquer, dynamic programming, greedy algorithms, graph algorithms, randomized algorithms, undecidability; NP-completeness, approximation algorithm, sorting lower bound, selected topics from amortized analysis, network flow, and linear programming. Three lecture hours and one hour of discussion a week for one semester. Only one of the following may be counted: [Computer Science 331](#), [331H](#), [378](#) (Topic: Algorithms and Complexity). Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [Mathematics 362K](#) or [Statistics and Data Sciences 321](#); and credit with a grade of at least C- or registration for: [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 333P. Science of High-Performance Computing.**

Explore aspects that impact high performance computing. Discuss "goal-oriented programming" that derives programs hand-in-hand with their proof of correctness. Examine techniques for mapping the appropriate algorithms to computer architectures so that they can attain high performance. Three lecture hours a week for one semester. [Computer Science 333P](#) and [378](#) (Topic: Prog Correctness/Performnc) may not both be counted. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#) and [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 234. Technical Writing.**

Application of techniques and strategies of effective technical writing, and of conventions used in documents such as letters, memos, proposals, abstracts, and reports. Two lecture hours a week for one semester. [Computer Science 234](#) and [178](#) (Topic: Technical Writing) may not both be counted. May not be counted toward the number of hours in computer science required for the Bachelor of Science in Computer Science degree. Prerequisite: [Computer Science 429](#), or [429H](#) with a grade of at least C-.

### **C S 337. Theory in Programming Practice.**

Application of program-analysis theory to program design. Methodologies for large-scale program design. Designed to help students bring together theoretical and programming skills. Three lecture hours and one discussion hour a week for one semester. [Computer Science 337](#) and [337H](#) may not both be counted. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); and [331](#), [331H](#), [341](#), or [341H](#).

### **C S 337H. Theory in Programming Practice: Honors.**

Application of program-analysis theory to program design. Methodologies for large-scale program design. Designed to help students bring together theoretical and programming skills. Three lecture hours and one discussion hour a week for one semester. [Computer Science 337](#) and [337H](#) may not both be counted. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); and [331](#), [331H](#), [341](#), or [341H](#).

### **C S 439. Principles of Computer Systems.**

Restricted to computer science majors. An introduction to computer systems software abstractions with an emphasis on the connection of these abstractions to underlying computer hardware. Key abstractions include threads, virtual memory, protection, and I/O. Requires writing of synchronized multithreaded programs and pieces of an operating system. Four lecture hours and two discussion hours a week for one semester. [Computer Science 439](#) and [439H](#) may not both be counted. Prerequisite: [Computer Science 429](#), or [429H](#) with a grade of at least C-.

### **C S 439H. Principles of Computer Systems: Honors.**

Restricted to computer science majors. An introduction to computer systems software abstractions with an emphasis on the connection of these abstractions to underlying computer hardware. Key abstractions include threads, virtual memory, protection, and I/O. Requires writing of synchronized multithreaded programs and pieces of an operating system. Four lecture hours and two discussion hours a week for one semester. [Computer Science 439](#) and [439H](#) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 340D. Debugging & Verifying Programs.**

Basic concepts of formal methods with emphasis on using and applying mathematical logic plus finite state systems theory to program verification and debugging. Three lecture hours a week for one semester. [Computer Science 340D](#) and [378](#) (Topic: Debugging and Verifying Programs) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 341. Automata Theory.**

An introduction to the formal study of automata and of related formal languages with applications in computer science. Three lecture hours a week for one semester. [Computer Science 341](#) and [341H](#) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 341H. Automata Theory: Honors.**

An introduction to the formal study of automata and of related formal languages with applications in computer science. Three lecture hours a week for one semester. [Computer Science 341](#) and [341H](#) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-, and consent of the honors director.

### **C S 342. Neural Networks.**

An introduction to deep learning covering basic building blocks and intuitions behind designing, training, and tuning of deep networks. Explore application areas including computer vision, sequence modeling in natural language processing, deep reinforcement learning, generative modeling, and adversarial learning. Three lecture hours a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 342C. Computational Brain.**

An examination of the brain as a computational system composed of different layers of abstraction. Focus on scientific writing, with weekly reviews of relevant scientific papers. Three lecture hours a week for one semester [Computer Science 342C](#) and [378](#) (Topic:



Computational Brain) may not both be counted. Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 429](#) or [429H](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### **C S 342S. Introduction to Speech and Audio Processing.**

Explore the theory and practice that underlies modern automatic speech recognition (ASR) systems and related technologies. Analyze the acoustic theory of speech production, digital signal processing for speech signals, Hidden Markov Models, Gaussian Mixture Models, language models, finite state transducers, decoding algorithms, and deep neural networks. Three lecture hours a week for one semester. [Computer Science 342S](#) and [378](#) (Topic: Intro to Speech and Audio Proces) may not both be counted. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [Mathematics 362K](#) or [Statistics and Data Sciences 321](#); and [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 343. Artificial Intelligence.**

An introduction to the field of artificial intelligence (AI), starting from foundations in search and knowledge representation, and then sampling from major AI topics such as probabilistic reasoning, intelligent agents (including for games), machine learning, planning, natural language processing, robotics, computer vision, and philosophical foundations. Programming projects are required. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 343](#), [343H](#), [378H](#) (Topic: Artificial Intelligence: Honors). Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 429](#) or [429H](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### **C S 343H. Artificial Intelligence: Honors.**

An introduction to the field of artificial intelligence (AI), starting from foundations in search and knowledge representation, and then sampling from major AI topics such as probabilistic reasoning, intelligent agents (including for games), machine learning, planning, natural language processing, robotics, computer vision, and philosophical foundations. Programming projects are required. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 343](#), [343H](#), [378H](#) (Topic: Artificial Intelligence: Honors). Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 429](#) or [429H](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### **C S 344M. Autonomous Multiagent Systems.**

An introduction to autonomous agents, with an emphasis on multiagent systems. Students use a robotics simulator. Emphasis on computer science research activities, including speaking, writing, programming, and working in groups. Three lecture hours a week for one semester. [Computer Science 344M](#) and [378](#) (Topic: Autonomous Multiagent Systems) may not both be counted. Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 429](#) or [429H](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### **C S 344R. Robotics.**

A survey of methods and techniques important for intelligent robotics. Students work in teams, applying these methods to get intelligent behavior from physical robots. Three lecture hours a week for one semester. [Computer Science 344R](#) and [378](#) (Topic: Robotics) may not both be counted. Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 429](#) or [429H](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### **C S 345. Programming Languages.**

Survey of significant concepts underlying modern programming languages, including syntax, functions, expressions, types, polymorphism, assignment, procedures, pointers, encapsulation, classes, and inheritance, with some discussion of implementation issues. Prominent programming paradigms, such as sequential, concurrent, object-oriented, functional, and logic programming. Illustrative examples drawn from a variety of current languages. Three lecture hours a week for one semester. [Computer Science 345](#) and [345H](#) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 345H. Programming Languages: Honors.**

A survey of significant concepts underlying modern programming languages, including syntax, functions, expressions, types, polymorphism, assignment, procedures, pointers, encapsulation, classes, and inheritance, with some discussion of implementation issues. Prominent programming paradigms, such as sequential, concurrent, object-oriented, functional, and logic programming. Illustrative examples drawn from a variety of current languages. Three lecture hours a week for one semester. [Computer Science 345](#) and [345H](#) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 346. Cryptography.**

A theoretical introduction to cryptography. Subjects include private key cryptosystems, public key cryptosystems, digital signatures, secret sharing schemes, and the necessary mathematical background. Three lecture hours a week for one semester. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [331](#), [331H](#), [341](#), or [341H](#); and credit with a grade of at least C- or registration for: [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 347. Data Management.**

Concepts of database design and database system implementation. Data models, query processing, database design theory, crash recovery, concurrent control, and distributed databases. Three lecture hours a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 349. Contemporary Issues in Computer Science.**

An examination of social, professional, and ethical issues involved in the use of computer technology. Subjects may include software engineering ethics, computer safety and reliability, constitutional issues, intellectual property, computer crime, societal impact, emerging technologies, and philosophical issues. Three lecture hours a week for one semester. [Computer Science 349](#) and [378](#) (Topic: Contemporary Issues in Computer Science) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 350C. Advanced Computer Architecture.**

Relationships between programming and hardware design, high performance and low power microarchitectures, memory system design, technology and workload constraints on design, special techniques for coding, addressing, and control, and evaluation of design alternatives will be explored. Three lecture hours a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### C S 350F. Operating Systems.

Advanced coverage of core operating systems (virtual memory, file systems, scheduling), and intermediate and advanced operating systems concepts: multiprocessing, multi-core computing, linking, kernel design, virtual machines, and security. Three lecture hours a week for one semester. Prerequisite: [Computer Science 439](#) or [439H](#) with a grade of at least C-.

### C S 353. Theory of Computation.

An introduction to computability theory and computational complexity, focusing on the limits of computation. Subjects include undecidability, NP-completeness, time and space complexity, circuit complexity, and randomized complexity. Three lecture hours a week for one semester. Prerequisite: One of the following with a grade of at least C-: [Computer Science 331](#), [331H](#), [341](#), or [341H](#).

### C S 354. Computer Graphics.

Introduction to techniques for human-machine communication through imagery. Subjects include display hardware, transformations, interactive techniques, geometric modeling, two- and three-dimensional display algorithms, graphics software systems architecture, and hidden-line and surface elimination. Projects are assigned and in-depth exploration is encouraged. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 354](#), [354H](#), [378H](#) (Topic: Computer Graphics Honors). Prerequisite: The following courses with a grade of at least a C-: [Computer Science 429](#) or [429H](#) and [Mathematics 340L](#) or [Statistics and Data Sciences 329C](#) or [Mathematics 341](#).

### C S 354H. Computer Graphics: Honors.

Introduction to techniques for human-machine communication through imagery. Explore display hardware, transformations, interactive techniques, geometric modeling, two- and three-dimensional display algorithms, graphics software systems architecture, and hidden-line and surface elimination. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 354](#), [354H](#), [378H](#) (Topic: Computer Graphics Honors). Prerequisite: The following courses with a grade of at least C-: [Computer Science 429](#) or [429H](#) and [Mathematics 340L](#) or [Statistics and Data Sciences 329C](#) or [Mathematics 341](#).

### C S 354P. Game Programming Paradigms.

Explore game programming paradigms in C++ using the Unreal Engine. Examine event-driven, component-oriented and object-oriented programming, as well as collaborative software development using source control, bug-tracking, and code documentation. Develop and apply skills towards building a game on the Unreal Engine. Three lecture hours a week for one semester. [Computer Science 354P](#) and [378](#) (Topic: Game Programming Paradigms) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least a C-.

### C S 354R. Game Technology.

An examination of major technological components of computer games and their integration, modeling and animation, physics simulation, networking for games, interface and interaction techniques, and special effects. Three lecture hours a week for one semester. [Computer Science 354R](#) and [378](#) (Topic: Game Technology) may not both be counted. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 105C](#), [371G](#) or [371P](#); and [Computer Science 439](#) or [439H](#).

### C S 354S. Game Development Capstone: Two-Dimensional Games.

Same as [Music 354S](#) and [Theatre and Dance 354T](#) (Topic 17). Participation in an interdisciplinary team to gain a thorough understanding of the two-dimensional game development process, through modeling of the environment and practices that are used in game studios. Three lecture hours a week for one semester. Only one of the following may be counted: [Arts and Entertainment Technologies 376](#), [Computer Science 354S](#), [378](#) (Topic: Game Devel Capstone: 2D Games), [Music 354S](#), [376J](#) (Topic: Game Devel Capstone: 2D Games), [Theatre and Dance 354T](#) (Topic 17), [Radio-Television-Film 344N](#) (Topic 6), [344T](#) (Topic 1). Prerequisite: For computer science majors, [Computer Science 354](#), [354H](#), [354P](#), or [354R](#) with a grade of at least C-; for others, consent of instructor.

### C S 354T. Game Development Capstone: Three-Dimensional Games.

Same as [Music 354T](#) and [Theatre and Dance 354T](#) (Topic 18). Participation in interdisciplinary teams to gain a thorough understanding of the three-dimensional game development process, through modeling of the environment and practices that are used in game studios. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 354T](#), [Music 354T](#), [376J](#) (Topic: Game Development Capstone: 3D Games), [Theatre and Dance 354T](#) (Topic 18). Prerequisite: For computer science majors, [Computer Science 354](#), [354H](#), [354P](#), or [354R](#) with a grade of at least C-; for others, consent of instructor.

### C S 356. Computer Networks.

Introduction to computer networks with emphasis on the Internet. Three lecture hours a week for one semester. Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 439](#) or [439H](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### C S 356R. Introduction to Wireless Networks.

An introduction to basic knowledge in wireless networks ranging from the physical layer to the application layer, and across different types of wireless networks such as WiFi, cellular, and multihop networks. Three lecture hours a week for one semester. [Computer Science 356R](#) and [378](#) (Topic: Introduction to Wireless Networks) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### C S 358H. Introduction to Quantum Information Science: Honors.

Introduction to the theory of quantum computing and information. Explore the rules of quantum mechanics (qubits, unitary transformations, measurements, density matrices); quantum gates and circuits; entanglement; the Bell inequality; protocols for teleportation, quantum key distribution, and other tasks; basic quantum algorithms such as Shor's and Grover's; basic quantum complexity theory; basic quantum error correction; decoherence and the measurement problem; and the challenges of building scalable quantum computers. Previous exposure to quantum mechanics is not required. Three lecture hours a week for one semester. [Computer Science 358H](#) and [378H](#) (Topic: Intro to Quantum Info Sci: Hon) may not both be counted. Prerequisite: [Computer Science 331](#) or [331H](#) with a grade of at least C-.

### C S 360V. Virtualization.

Examine the latest in virtualization technologies such as virtual machines, containers and serverless computing. Three lecture hours a week for one semester. [Computer Science 360V](#) and [378](#) (Topic: Virtualization) may not both be counted. Prerequisite: [Computer](#)

[Science 439](#) or [439H](#) with a grade of at least C-.

### **C S 361. Introduction to Computer Security.**

An introduction to computer security, both in the abstract and in the context of real systems, including recognizing potential threats to confidentiality, integrity and availability, and developing familiarity with current security-related issues in computer science. Three lecture hours a week for one semester. [Computer Science 361](#) and [378](#) (Topic: Introduction to Security) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 361C. Information Assurance and Security.**

Selected subjects from the broad area of information assurance, including computer security, communications security, operations security, and risk assessment. Three lecture hours a week for one semester. [Computer Science 361C](#) and [378](#) (Topic: Information Assurance and Security) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 361S. Network Security and Privacy.**

An examination of common computer security threats; fundamental techniques for implementing secure software systems and networks; practical experience in attacking and defending networked systems. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 361S](#), [378](#) (Topic: Network Security and Privacy), [378H](#) (Topic: Network Security/Privacy:Hon). Prerequisite: [Computer Science 439](#) or [439H](#) with a grade of at least C-.

### **C S 363H. Principles of Machine Learning I: Honors.**

Introduction to machine learning: data processing, regression, classification, clustering, dimensionality reduction, and neural networks. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 363D](#), [363H](#), [363M](#), [378](#) (Topic: Introduction to Data Mining), [378H](#) (Topic: Intro to Data Mining: Honors). Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [Mathematics 362K](#) or [Statistics and Data Sciences 321](#); and [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 363M. Principles of Machine Learning I.**

Introduction to machine learning: data processing, regression, classification, clustering, dimensionality reduction, and neural networks. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 363D](#), [363H](#), [363M](#), [378](#) (Topic: Introduction to Data Mining), [378H](#) (Topic: Intro to Data Mining: Honors). Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [Mathematics 362K](#) or [Statistics and Data Sciences 321](#); and [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 364M. Principles of Machine Learning II.**

Introduction to advanced machine learning: optimization methods, statistical inference and probabilistic modeling, generalization, and uncertainty quantification. Three lecture hours a week for one semester. [Computer Science 364D](#) and [364M](#) may not both be counted. Prerequisite: Upper-division standing and [Computer Science 363D](#), [363M](#) or [363H](#), and [331](#) or [331H](#) with a grade of at least C-.

### **C S 365T. Trustworthy Machine Learning.**

Explore ways in which contemporary machine learning (ML) algorithms can be untrustworthy, along with algorithmic approaches to mitigate these trust issues. Discuss algorithmic fairness, privacy in ML, ML security, and interpretability/explicability in ML. Examine fairness, security, and privacy issues in ML. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 365T](#), [378](#) (Topic: Trustworthy Machine Learning), [378](#) (Topic: Safe and Ethical AI). Prerequisite: [Computer Science 331](#) or [331H](#) with a grade of at least C-.

### **C S 367. Numerical Methods.**

Subjects include systems of linear equations, numerical integration, ordinary differential equations, and nonlinear equations. Construction and use of large numerical systems. Influence of data representation and computer architecture on algorithm choice and development. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 367](#), [Mathematics 368K](#), [Physics 329](#). Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [Mathematics 408D](#), [408M](#), or [427L](#); and [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#).

### **C S 369. Systems Modeling.**

Introduction to performance modeling, with emphasis on computer systems. Modeling methodology, queueing network models, simulation, analysis of results. Three lecture hours a week for one semester. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### **C S 370. Undergraduate Reading and Research.**

Supervised study of selected problems in computer science, by individual arrangement with supervising instructor. The equivalent of three lecture hours a week for one semester. [Computer Science 370](#) and [370F](#) may not both be counted. No more than three semester hours may be counted toward a degree in computer science. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-, and consent of the undergraduate adviser.

### **C S 370F. Undergraduate Reading and Research: Writing.**

Supervised study of selected problems in computer science, by individual arrangement with supervising instructor. The equivalent of three lecture hours a week for one semester. [Computer Science 370](#) and [370F](#) may not both be counted. No more than three semester hours may be counted toward a degree in computer science. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-, and consent of the undergraduate adviser.

### **C S 371D. Distributed Computing.**

Investigation of models, principles, and fundamental protocols, including event ordering and global predicate detection, atomic commit, state-machine replication, rollback recovery, primary backup, consensus for synchronous and asynchronous systems, and byzantine fault-tolerance. Three lecture hours a week for one semester. Prerequisite: [Computer Science 439](#) or [439H](#) with a grade of at least C-.

### **C S 371G. Generic Programming and the STL.**

Three lecture hours a week for one semester. [Computer Science 371G](#) and [378](#) (Topic: Generic Programming and the STL) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 371L. iOS Mobile Computing.**

Overview of mobile computing with an emphasis on developing applications for the iOS operating system. Three lecture hours a week for one semester. [Computer Science 371L](#) and [378](#) (Topic: iOS Mobile Computing) may not both be counted. Prerequisite: Credit with a grade of at least C- or registration in [Computer Science 439](#) or [439H](#).

### **C S 371M. Mobile Computing.**

Overview of mobile computing. Emphasis on mobile application development. Use of mobile capabilities: sensors, location, telephony, and multimedia. Three lecture hours a week for one semester [Computer Science 371M](#) and [378](#) (Topic: Mobile Computing) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-; and credit with a grade of at least C- or registration for [Computer Science 439](#) or [439H](#).

### **C S 371N. Natural Language Processing.**

Explore modern natural language processing using machine learning and deep learning approaches. Examine linguistics fundamentals (syntax, semantics, distributional properties of language), machine learning models (classifiers, sequence taggers, deep neural networks), key algorithms for inference, and a range of applications. Build systems to do tasks including sentiment classification, text generation, and more. Three lecture hours a week for one semester. [Computer Science 371N](#) and [378](#) (Topic: Natural Language Processing) may not both be counted. Prerequisite: [Computer Science 331](#) or [331H](#) with a grade of at least a C-.

### **C S 371P. Object-Oriented Programming.**

Programming using class derivation, inheritance, and dynamic polymorphism. Application of a simple object-oriented design methodology to several software development problems. Three lecture hours a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 371R. Information Retrieval and Web Search.**

Introduction to traditional and recent methodologies for indexing, processing, querying, and classifying unstructured and semi-structured textual data, including hypertext and World Wide Web documents. Three lecture hours a week for one semester. [Computer Science 371R](#) and [378](#) (Topic: Intelligent Information Retrieval and Web Search) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 371S. Object-Oriented Software Engineering.**

Object-oriented formulations of software systems as executable specifications, object-oriented analysis, design of software architectures, translation of high-level specification systems. Three lecture hours a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 373. Software Engineering.**

Introduction to current knowledge, techniques, and theories in large software system design and development. Three lecture hours a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 173G. Social Impact Capstone.**

Supervised study of social-impact related problems in computer science. One lecture hour a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C- and consent of instructor.

### **C S 373S. Software Design.**

Model-driven engineering, UML metamodels and constraints, model transformations, object-oriented refactorings, design patterns, object-oriented frameworks and plug-ins, layered and dataflow architectural styles, map-reduce parallel software architectures, and service-oriented architectures. Three lecture hours a week for one semester [Computer Science 373S](#) and [378](#) (Topic: Software Design) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 374L. Longhorn Startup.**

Focus on skill development and mentoring in startup formation, technology development, market validation, marketing, sales, operations, human resources, program management, and finance. Emphasis on written and oral presentation of startup activities. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 374L](#), [378](#) (Topic: Longhorn Startup), [Electrical and Computer Engineering 364E](#), Electrical Engineering [377E](#), [Engineering Studies 377E](#), [Management 337](#) (Topic 2), [372T](#) (Topic 2). Prerequisite: Upper-division standing, [Computer Science 429](#) or [429H](#) with a grade of at least C-, and consent of instructor.

### **C S 174S. Longhorn Startup Seminar.**

An introduction to the world of startups through the experiences of people who have built successes and learned from failures. Two lecture hours a week for one semester. Only one of the following may be counted: [Biomedical Engineering 177T](#), [277T](#), [377T](#) (Topic: Longhorn Startup Seminar), [Computer Science 178](#), [174S](#), [378](#) (Topic: Longhorn Startup Seminar), [Engineering Studies 177](#), [277](#), [377](#) (Topic: Longhorn Startup Seminar), [177](#), [277](#), [377](#) (Topic 1), [Fine Arts 160](#), [260](#), [360](#) (Topic: Longhorn Startup Seminar), [170](#), [270](#), [370](#), [Management 137](#), [237](#), [337](#) (Topic: Longhorn Startup Seminar), [Communication 125](#), [225](#), [325](#) (Topic: Longhorn Startup Seminar). Prerequisite: Upper-division standing, [Computer Science 429](#) or [429H](#) with a grade of at least C-, and consent of instructor.

### **C S 375. Compilers.**

An examination of formal description of languages, lexical analysis, syntax analysis, syntax-directed translation, run-time system management, code generation, code optimization, compiler-building tools. Three lecture hours a week for one semester. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 376. Computer Vision.**

Explores computer vision, a discipline that develops methods that enable machines to interpret or analyze images and videos. Includes the study of image formation, feature detection, segmentation, multiple-view geometry, recognition and learning, and motion and tracking. Three lecture hours a week for one semester. [Computer Science 376](#) and [378](#) (Topic: Computer Vision) may not both be



counted. Prerequisite: The following coursework with a grade of at least C-: [Computer Science 429](#) or [429H](#); [Mathematics 340L](#), [341](#), or [Statistics and Data Sciences 329C](#); and [Mathematics 362K](#) or [Statistics and Data Sciences 321](#).

### **C S 377. Principles and Applications of Parallel Programming.**

Models of parallel computation, fundamental concepts for representation of parallel computation structures, study of representative parallel programming systems, and programming of parallel algorithms and computations. Three lecture hours a week for one semester. Prerequisite: [Computer Science 439](#) or [439H](#) with a grade of at least C-.

### **C S 377P. Programming for Performance.**

Investigation on the performance-critical features of modern computer architectures and how applications can take advantage of them to obtain high performance. Three lecture hours a week for one semester. [Computer Science 377P](#) and [378](#) (Topic: Programming for Performance) may not both be counted. Prerequisite: [Computer Science 429](#) or [429H](#) with a grade of at least C-.

### **C S 178, 378. Undergraduate Topics in Computer Science.**

One or three lecture hours a week for one semester. Only one of the following may be counted unless the topics vary: [Computer Science 178](#), [378](#), [178H](#), [378H](#). May be repeated for credit when the topics vary. Prerequisite: Upper-division standing; additional prerequisites vary with the topic.

**Topic 1: Quantum Computing II.** Introduction to quantum algorithms and programming. Computer Science 378 (Topic: Quantum Computing-FRI) and 378 (Topic 1) may not both be counted.

### **C S 178H, 378H. Undergraduate Topics in Computer Science: Honors.**

For each semester hour of credit earned, one lecture hour a week for one semester. Only one of the following may be counted unless the topics vary: [Computer Science 178](#), [378](#), [178H](#), [378H](#). May be repeated for credit when the topics vary. Prerequisite: [Computer Science 429](#) (or [310](#)) or [429H](#) (or [310H](#)) with a grade of at least C-.

### **C S 379H. Computer Science Honors Thesis.**

Directed reading, research, and/or projects in areas of computer science, under supervision of a faculty member, leading to an honors thesis. The thesis must be approved by a committee of three readers. The equivalent of three lecture hours a week for one semester, by arrangement with a faculty member. Prerequisite: The following coursework with a grade of at least C- in each: [Computer Science 311](#), or [311H](#); [Computer Science 314](#), or [314H](#); [Computer Science 429](#), or [429H](#); [Mathematics 408C](#), [408K](#), or [408N](#); nine additional semester hours of upper-division coursework in computer science; and consent of the student's research supervisor and the departmental honors adviser.

## **Graduate Courses**

### **C S 380C. Compilers.**

Basics of static analysis and transformation techniques; exploration in depth of one aspect of compilation and optimization. Three lecture hours a week for one semester. [Computer Science 380C](#) and [395T](#) (Topic: Compilers) may not both be counted. Prerequisite: Graduate standing; [Computer Science 375](#) are recommended.

### **C S 380D. Distributed Computing I.**

Models of distributed systems; language issues, proving properties of distributed systems; time, clocks, partial ordering of events; deadlock and termination detection; diffusing computations; computing in hostile environments; distributed resource management. Three lecture hours a week for one semester. Prerequisite: Graduate standing.

### **C S 380I. Structure and Implementation of Modern Programming Languages.**

Explore the breadth of these technologies for translating high-level program text to binary machine code within a unified cost-benefit-risk analysis framework. Illustrate system integration by building a series of compilers for LiveOak, a pedagogical, object-oriented language. Discuss lexical analysis, syntactic analysis, code generation, register allocation, language-level virtual machines, dynamic memory management, linking, and interoperability. Three lecture hours a week for one semester. [Computer Science 380I](#) and [395T](#) (Topic: SIMPL) may not both be counted. Prerequisite: Graduate standing; and extensive experience programming in Java emphasizing the use, augmentation, refactoring, and integration of pre-existing components; prior courses in programming languages, assembly language, compilers, computer architecture, and automata theory are helpful but not required.

### **C S 380J. Operating Systems Implementation.**

Seminar about how operating systems work. Readings and discussion about classic and recent research papers. Intensive programming assignments in the construction of a prototype operating system. Three lecture hours a week for one semester. [Computer Science 380J](#) and [395T](#) (Topic: Operating Systems Implementation) may not both be counted. Prerequisite: Graduate standing; and an undergraduate course in operating systems, networking, or distributed systems, or consent of instructor.

### **C S 380L. Advanced Operating Systems.**

Study of the formal structure, design principles, organization, implementation, and performance analysis of multiprogramming and/or multiprocessor computer systems. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and consent of instructor.

### **C S 380N. Systems Modeling.**

Theory and applications of Markovian models: birth-death models, queueing models, and networks of queues. Numerical methods: computational algorithms, approximation techniques, discrete-event simulation. Performance of scheduling disciplines: priority, time-sharing, multiple access. Three lecture hours a week for one semester. Prerequisite: Graduate standing and an undergraduate course in probability theory.

### **C S 380P. Parallel Systems.**

Explores parallel systems, from languages to hardware, from large-scale parallel computers to multicore chips, and from traditional parallel scientific computing to modern uses of parallelism. Includes discussion of and research methods in graphics, languages, compilers, architecture, and scientific computing. Three lecture hours a week for one semester. [Computer Science 380P](#) and [395T](#) (Topic: Parallel Systems) may not both be counted. Prerequisite: Graduate standing.

### **C S 380S. Theory and Practice of Secure Systems.**

Survey of modern security, designed to introduce the basic techniques used in the design and analysis of secure systems. Three lecture hours a week for one semester. Prerequisite: Graduate standing and prior undergraduate coursework in computer architecture and operating systems.

### **C S 380V. Virtualization.**

Discuss the latest in virtualization technologies, such as virtual machines, containers, serverless, etc. Explore CPU virtualization, memory virtualization, networking virtualization, storage virtualization, paravirtualization, containers, unikernels, and serverless computing. Three lecture hours a week for one semester. [Computer Science 380V](#) and [395T](#) (Topic: Virtualization) may not both be counted. Prerequisite: Graduate standing and coursework in operating systems and computer architecture.

### **C S 381K. Artificial Intelligence.**

Use of computers in problem solving, game playing, theorem proving, natural language understanding, and related tasks; methods of search, knowledge representation, learning, and other topics. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and [Computer Science 351](#) or consent of instructor.

### **C S 381V. Visual Recognition.**

Subjects include fundamental representations, learning approaches, matching-based algorithms, human activity models for video, and large-scale recognition. Three lecture hours a week for one semester. [Computer Science 381V](#) and [395T](#) (Topic: Visual Recognition) may not both be counted. Prerequisite: Graduate standing; a background in basic computer vision and machine learning is recommended.

### **C S 382M. Advanced Computer Architecture.**

Algorithms and their realizations, special techniques for coding, addressing, and control; integration of computer units; relations between programming and design considerations. Three lecture hours a week for one semester. Prerequisite: Graduate standing.

### **C S 383C. Numerical Analysis: Linear Algebra.**

Same as [Computational Science, Engineering, and Mathematics 383C](#), [Mathematics 383E](#), and [Statistics and Data Sciences 393C](#). Survey of numerical methods in linear algebra: floating-point computation, solution of linear equations, least squares problems, algebraic eigenvalue problems. Three lecture hours a week for one semester. Only one of the following may be counted: [Computational Science, Engineering, and Mathematics 383C](#), [Computer Science 383C](#), [Mathematics 383E](#), [Statistics and Data Sciences 393C](#). Prerequisite: Graduate standing; [Computer Science 367](#) or [Mathematics 368K](#); and [Mathematics 340L](#), [341](#), or consent of instructor.

### **C S 383D. Numerical Analysis: Interpolation, Approximation, Quadrature, and Differential Equations.**

Same as [Computational Science, Engineering, and Mathematics 383D](#), [Mathematics 383F](#), and [Statistics and Data Sciences 393D](#). Survey of numerical methods for interpolation, functional approximation, integration, and solution of differential equations. Three lecture hours a week for one semester. Only one of the following may be counted: [Computational Science, Engineering, and Mathematics 383D](#), [Computer Science 383D](#), [Mathematics 383F](#), [Statistics and Data Sciences 393D](#). Prerequisite: Graduate standing; [Computational Science, Engineering, and Mathematics 383C](#), [Computer Science 383C](#), [Mathematics 383E](#), or [Statistics and Data Sciences 393C](#); and [Mathematics 427K](#) and [365C](#), or consent of instructor.

### **C S 384G. Computer Graphics.**

Same as [Computational Science, Engineering, and Mathematics 382G](#). Advanced material in computer graphics, including in-depth treatments of techniques for realistic image synthesis, advanced geometric modeling methods, animation and dynamic simulation, scientific visualization, and high-performance graphics architectures. Three lecture hours a week for one semester. [Computational Science, Engineering, and Mathematics 382G](#) and [Computer Science 384G](#) may not both be counted. Prerequisite: Graduate standing; and [Computer Science 354](#) or another introductory course in computer graphics, or equivalent background and consent of instructor.

### **C S 384M. Multimedia Systems.**

Theoretical and practical issues in advanced systems, including multimedia systems, digital audio and video compression techniques, operating system and network support for digital audio and video, and multimedia conferencing systems. Three lecture hours a week for one semester. Prerequisite: Graduate standing; and either [Computer Science 356](#) or [380D](#) and [380L](#).

### **C S 384P. Physical Simulation of Computer Graphics.**

Explore the key concepts and algorithms for simulating physical systems; starting from the ground up with particle systems and mass-spring networks, rigid and elastic bodies, collisions, cloth, and fluids. Three lecture hours a week for one semester. [Computer Science 384P](#) and [395T](#) (Topic: Physical Simulation) may not both be counted. Prerequisite: Graduate standing; and an introductory course in computer graphics, or equivalent background and consent of instructor.

### **C S 384R. Geometric Modeling and Visualization.**

Computational image processing, computational geometry and geometric modeling algorithms with an emphasis on spatial realism, and the programmatic use of physiological simulation and visualization to quantitatively depict how things work at the molecular, cellular, tissue, organ, and system levels. Three lecture hours a week for one semester. [Computer Science 384R](#) and [395T](#) (Topic: Graphics, Modeling, and Visualization) may not both be counted; [Computer Science 384R](#) and [395T](#) (Topic: Multiscale Bio-Modeling and Visualization) may not both be counted; [Computer Science 384R](#) and [395T](#) (Topic: Physically Based Geometric Modeling) may not both be counted. Prerequisite: Graduate standing, and [Computer Science 354](#) or consent of instructor.

### **C S 384V. Introduction to VLSI Design.**

Basic techniques required to design custom negative metal oxide semiconductor digital integrated circuits. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and consent of instructor.

### **C S 386C. Dependable Computing Systems.**

System models from synchronous to asynchronous, with emphasis on in-between models such as the timed asynchronous model. Control structures such as timed state-transition systems, and constraints in temporal and real-time logics. Analysis techniques such

as model checking of timed systems, and extended Presburger arithmetic. Basic building blocks such as clock synchronization, synchronous atomic broadcast, time-bounded membership protocols, real-time scheduling theory, and state recovery methods. Practical implementation issues such as special operating system data structures and algorithms, open system design, and security concerns. Three lecture hours a week for one semester. [Computer Science 386C](#) and [395T](#) (Topic: Dependable Computing Systems) may not both be counted. Prerequisite: Graduate standing, and an undergraduate course in operating systems or consent of instructor.

### **C S 386D. Database Systems.**

Introduction to the principles of database systems, including fundamental ideas and algorithms used in the construction of centralized database management systems, distributed database management systems, and database machines and their roles in Internet infrastructure. Topics include data storage and indexing algorithms, query processing and optimization, concurrency control, recovery, XML and object-oriented databases, database evaluation and tuning, and recent directions in database research. Three lecture hours a week for one semester. Prerequisite: Graduate standing and [Computer Science 347](#) and [375](#).

### **C S 386K. Numerical Treatment of Differential Equations.**

The analysis of numerical methods for solving ordinary and partial differential equations. Three lecture hours a week for one semester. Prerequisite: Graduate standing; and , [Computer Science 383D](#), [Mathematics 368K](#), [383F](#), or consent of instructor.

### **C S 386L. Programming Languages.**

Topics include formal syntax representations, program correctness, typing, and data abstraction. Features and problems in languages that allow parallelism. Exploration of different programming styles, such as imperative, functional, logic, data flow, and object-oriented programming. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and [Computer Science 345](#) or consent of instructor.

### **C S 386M. Communication Networks.**

Switching techniques, network and protocol architectures, communication protocols, resource allocation problems, internetworking, design and analysis methods. Three lecture hours a week for one semester. Prerequisite: Graduate standing.

### **C S 386P. Android Programming.**

Study Android APIs and learn to build significant Android applications. Explore general principles of software engineering and mobile app development. Three lecture hours a week for one semester. [Computer Science 386P](#) and [395T](#) (Topic: Android Programming) may not both be counted. Prerequisite: Graduate standing and knowledge of programming, object-oriented terminology, and Java.

### **C S 386S. Network Protocol Security.**

Techniques and research in Internet and network security. Three lecture hours a week for one semester. [Computer Science 386S](#) and [395T](#) (Topic: Secure Network Protocols) may not both be counted. Prerequisite: Graduate standing.

### **C S 386W. Wireless Networking.**

Fundamental concepts and principles of wireless network technologies and protocol design, ranging from physical layer to application layer, and in-depth studies of current wireless research. Three lecture hours a week for one semester. [Computer Science 386W](#) and [395T](#) (Topic: Wireless Networking) may not both be counted. Prerequisite: Graduate standing.

### **C S 388. Natural Language Processing.**

Computational methods for syntactic and semantic analysis of structures representing meanings of natural language; study of current natural language processing systems; methods for computing outlines and discourse structures of descriptive text. Three lecture hours a week for one semester. Only one of the following may be counted: [Artificial Intelligence 388](#), [Computer Science 388](#), [Data Science 388](#), [395T](#) (topic: Natural Language Processing) Prerequisite: Graduate standing, and a course in artificial intelligence or consent of instructor.

### **C S 388C. Combinatorics and Graph Theory.**

Counting, matching theory, extremal set theory, Ramsey theory, probabilistic method, linear algebra method, coding theory. Applications to computer science, including randomized algorithms. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and consent of instructor. An understanding of elementary proof and counting techniques is assumed.

### **C S 388D. Data-Driven Algorithm Design.**

Analyze recent developments on the role of machine learning in designing algorithms from the dual viewpoints of tailoring algorithm design to exploit the availability of data as well as tailoring learning to optimize for algorithm performance. Three hours a week for one semester [Computer Science 388D](#) and [395T](#) (Topic: Data-Driven Algorithm Design) may not both be counted. Prerequisite: Graduate standing and knowledge of undergraduate algorithms and complexity, and at least one graduate course in theoretical computer science. Prior knowledge in randomized algorithms and machine learning.

### **C S 388E. Approximation Algorithms and Complexity.**

Explore approximation algorithms for NP-hard problems, online algorithms, approximation in P, and other subjects related to approximation algorithms and their limitations. Examine combinatorial algorithms, as well as algorithms based on linear programming and semidefinite programming. Identify the Probabilistically Checkable Proofs (PCP) Theorem, discuss its applications, and prove a weak version of it. Three lecture hours a week for one semester. [Computer Science 388E](#) and [395T](#) (Topic: Approximtn Algorithms/Complexity) may not both be counted. Prerequisite: Graduate standing and knowledge of undergraduate-level algorithms and complexity.

### **C S 388F. Automata and Formal Languages.**

Formal grammars, languages and related classes of automata, language hierarchies, operations on languages, decidability, related complexity issues, closure properties, other classes of automata. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and consent of instructor.

### **C S 388G. Algorithms: Techniques and Theory.**

Explore advanced subjects in algorithm design and analysis including algorithmic paradigms, data structures, maximum flow, randomized algorithms, NP-completeness, and approximation algorithms. Three lecture hours a week for one semester. [Data Science 388G](#) and [Computer Science 388G](#) may not both be counted. Prerequisite: Graduate standing and knowledge of algorithmic paradigms and discrete mathematics.

### **C S 388H. Cryptography.**

Surveys the foundations of cryptography from formal notions of security to fundamental protocols, including one-way functions, encryption, pseudorandom generators, signature schemes, and zero-knowledge. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and [Computer Science 353](#) or consent of instructor.

### **C S 388J. Optimization.**

Explore the background on convex sets and functions, linear programming, convex programming, and iterative first-order and second-order methods. Three lecture hours a week for one semester. Only one of the following may be counted: [Artificial Intelligence 388J](#), [Computer Science 388J](#), [395T](#) (Topic: Optimization). Prerequisite: Graduate standing.

### **C S 388K. Online Learning and Optimization.**

Explore algorithms for convex optimization and algorithms for online learning. Focus on algorithms for large scale convex optimization. Discuss problems in machine learning. Apply these ideas to online learning. Three lecture hours a week for one semester. Only one of the following may be counted: [Artificial Intelligence 388K](#), [Computer Science 388K](#), [395T](#) (Topic: Online Learning/Optimization). Prerequisite: Graduate standing.

### **C S 388L. Introduction to Mathematical Logic.**

Introduction to some of the principal topics of mathematical logic: propositional and predicate calculus; Goedel's completeness theorem; first-order theories; formalizing mathematical reasoning; first-order arithmetic; recursive functions; Goedel's incompleteness theorems; axiomatic set theory. Three lecture hours a week for one semester. Prerequisite: Graduate standing and experience in abstract mathematical thinking.

### **C S 388M. Communication Complexity.**

Covers the most important models of communication complexity and their applications, including recent research results and various open problems. Three lecture hours a week for one semester. [Computer Science 388M](#) and [395T](#) (Topic: Communication Complexity) may not both be counted. Prerequisite: Graduate standing.

### **C S 388P. Parallel Algorithms.**

Theory of parallel computation: Parallel algorithm design and analysis on shared memory machines; parallel complexity; other models of parallel computation; lower bounds. Three lecture hours a week for one semester. Prerequisite: Graduate standing; and [Computer Science 388G](#), or consent of instructor.

### **C S 388Q. Quantum Information Science.**

Introduction to the theory of quantum computing and information. Examine the rules of quantum mechanics (qubits, unitary transformations, density matrices); quantum gates and circuits; entanglement; the Bell inequality; protocols for teleportation, quantum key distribution, quantum money and other tasks; basic quantum algorithms such as Shor's and Grover's; basic quantum complexity theory; and the challenges of building scalable quantum computers. Three lecture hours a week for one semester. [Computer Science 388Q](#) and [395T](#) (Topic: Quantum Information Science) may not both be counted. Prerequisite: Graduate standing; and coursework in linear algebra and some exposure to classical algorithms and programming; and an understanding of complex numbers and linear algebra: vector spaces, bases, eigenvalues, rank, inner product, etc; previous exposure to quantum mechanics is not required.

### **C S 388R. Randomized Algorithms.**

The design and analysis of efficient randomized algorithms. Three lecture hours a week for one semester. [Computer Science 388R](#) and [395T](#) (Topic: Randomized Algorithms) may not both be counted. Prerequisite: Graduate standing.

### **C S 388S. Formal Semantics and Verification.**

Sequential execution: partial and total correctness; deductive, operational, and denotational semantics; formal derivation of programs; parallel execution: partial correctness, deadlock, and starvation; methodology, parallel versus distributed execution. Three lecture hours a week for one semester. Prerequisite: Graduate standing and consent of instructor.

### **C S 388T. Theory of Computation.**

Examine polynomial hierarchy, circuit lower bounds, randomized computation, approximability, communication complexity, pseudorandom generators, and probabilistically checkable proofs. Three lecture hours a week for one semester. Prerequisite: Graduate standing; and a course in computational complexity or theory of computation, or equivalent background and consent of instructor.

### **C S 388U. Planning, Search, and Reasoning Under Uncertainty.**

Introduction to three key foundational problems in AI: planning, search, and reasoning under uncertainty. Investigate how to define planning domains, including representations for world states and actions, covering both symbolic and path planning. Study algorithms to efficiently find valid plans with or without optimality, and partially ordered, or fully specified solutions. Three hours a week for one semester. Only one of the following may be counted: [Artificial Intelligence 388U](#), [Computer Science 388U](#), [395T](#) (Topic: Planning Search and Reasoning). Prerequisite: Graduate standing; and knowledge of undergraduate data structures, algorithms, and complexity; and prior knowledge in probability theory or statistics.

### **C S 389L. Automated Logical Reasoning.**

Subjects include automated reasoning techniques for propositional logic, first-order logic, linear arithmetic over reals and integers, theory of uninterpreted functions, and combinations of these theories. Examines automated logical reasoning both from a theoretical and practical perspective, giving a hands-on experience building useful tools, such as SAT and SMT solvers. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 389L](#), [395T](#) (Topic: Automated Logical Reasoning), [Artificial Intelligence 389L](#). Prerequisite: Graduate standing.



### **C S 389M. Principles of Object-Oriented Software Technology.**

Fundamental principles of object-oriented software engineering, including design and implementation of object-oriented analysis methods, software architectures, translators of high-level programming language representations, translations to multiple-software architectures. Three lecture hours a week for one semester. Prerequisite: Graduate standing, [Computer Science 371S](#) or the equivalent, and consent of instructor.

### **C S 389R. Recursion and Induction I.**

The development of a formal theory for reasoning about computer programs, with emphasis on recursively defined functions in the LISP style and proof by mathematical induction. Heavy emphasis on student discovery and presentation of proofs. Three lecture hours a week for one semester. Prerequisite: Graduate standing.

### **C S 390D. Distributed Computing II.**

Synchronous and asynchronous algorithms, with particular emphasis on notations for expressing the algorithms and logics for reasoning about them. Algorithms from a variety of application areas and for a variety of architectures. Three lecture hours a week for one semester. Prerequisite: Graduate standing and [Computer Science 380D](#).

### **C S 390S. Sublinear Algorithms.**

Study algorithms that can process very large data sets, with a focus on streaming algorithms, compressed sensing, and property testing. Three lecture hours a week for one semester. [Computer Science 390S](#) and [395T](#) (Topic: Sublinear Algorithms) may not both be counted. Prerequisite: Graduate standing and knowledge of algorithms, probability, and linear algebra.

### **C S 391D. Data Mining: A Mathematical Perspective.**

Mathematical and statistical aspects of data mining. Topics include supervised learning (regression, classification, support vector machines) and unsupervised learning (clustering, principal components analysis, dimensionality reduction). Uses technical tools that draw from linear algebra, multivariate statistics, and optimization. Three lecture hours a week for one semester. [Computer Science 391D](#) and [395T](#) (Topic: Data Mining: A Statistical Learning Perspective) may not both be counted. Prerequisite: Graduate standing, and [Mathematics 341](#) or the equivalent.

### **C S 391K. Artificial Intelligence II.**

Advanced course in artificial intelligence. Topics include planning, probabilistic reasoning, truth maintenance, abduction, model-based diagnosis, and speech recognition. Three lecture hours a week for one semester. Prerequisite: Graduate standing, and [Computer Science 381K](#) or equivalent knowledge of artificial intelligence and LISP.

### **C S 391L. Machine Learning.**

Computing systems that automatically improve their performance with experience, including various approaches to inductive classification such as version space, decision tree, rule-based, neural network, Bayesian, and instance-based methods; as well as computational learning theory, explanation-based learning, and knowledge refinement. Three lecture hours a week for one semester. Only one of the following may be counted: [Artificial Intelligence 391L](#), [Computer Science 391L](#), [Data Science 391L](#). Prerequisite: Graduate standing, and [Computer Science 381K](#) or equivalent knowledge of artificial intelligence and LISP.

### **C S 391R. Robot Learning.**

Survey a wide range of modern techniques in robotics that learn from data, largely focusing on robot perception and decision making. Explore 3D vision, representation learning, active perception, reinforcement learning, imitation learning, and applications in robot manipulation. Three lecture hours a week for one semester. [Computer Science 391R](#) and [395T](#) (Topic: Robot Learning) may not both be counted. Prerequisite: Graduate standing, and experience in machine learning and computer vision.

### **C S 392C. Methods and Techniques for Parallel Programming.**

Models of parallel fundamental concepts for representation of parallel computation structures, study of representative parallel programming languages, formulation of languages and translation methods, translation of parallel programs to multiple targets, laboratory exercises in parallel programming. Three lecture hours a week for one semester. Prerequisite: Graduate standing and consent of instructor.

### **C S 392F. Automated Software Design.**

Model-driven engineering; UML metamodels and constraints, model transformations, software product lines, feature models, feature modularity, feature algebras, feature interactions, multi-dimensional separation of concerns, design-by-transformation, parallel software architectures, correct-by-construction, architecture refinement, optimization, and extension, program refactorings, design patterns, refactoring scripts, category theory, functors, commuting diagrams. Three lecture hours a week for one semester. [Computer Science 392F](#) and [395T](#) (Topic: Feature-Oriented Programming) may not both be counted. Prerequisite: Graduate standing, and a basic knowledge of Java, compilers and grammars, and object-oriented design methods.

### **C S 393C. Agent-Based Electronic Commerce.**

Focuses on the intersection of computer science (including multiagent systems and machine learning), economics, and game theory. Explores economic mechanisms of exchange suitable for use by automated intelligent agents, including auctions and auction theory, game theory and mechanism design, and autonomous bidding agents. Students demonstrate programming proficiency in a trading agent competition. Three lecture hours a week for one semester. [Computer Science 393C](#) and [395T](#) (Topic: Agent-Based Electronic Commerce) may not both be counted. Prerequisite: Graduate standing.

### **C S 393D. Topics in Numerical Analysis.**

Recent topics have included numerical methods in ordinary differential equations, numerical methods in partial differential equations, computational problems in linear algebra, numerical solution of systems of equations, numerical methods in functional approximation, numerical integration. Three lecture hours a week for one semester. May be repeated for credit when the topics vary. Prerequisite: Graduate standing and consent of instructor.

### **C S 393N. Numerical Solution of Elliptic Partial Differential Equations.**

Same as [Mathematics 393N](#). The numerical solution of large systems of linear algebraic equations arising in the solution of elliptic partial differential equations by discretization methods. Three lecture hours a week for one semester. Computational Science,

Engineering and [Mathematics 393N](#) and [Mathematics 393N](#) may not both be counted. Prerequisite: Graduate standing; and [Computational Science, Engineering, and Mathematics 383K](#), [Computer Science 386K](#), [Mathematics 387C](#), or consent of instructor.

### **C S 393P. Program Synthesis.**

Study the methods for automatically discovering programs that implement given high-level specifications, which can take many forms, including formal constraints, input-output examples, demonstrations, and natural language. Examine contemporary algorithms for program synthesis to bring together symbolic techniques for pruning and search, numerical techniques for finding optimal parameters, and statistical techniques for interpreting ambiguous specifications and learning to search. Three lecture hours a week for one semester. [Computer Science 393P](#) and [395T](#) (Topic: Program Synthesis) may not both be counted. Prerequisite: Graduate standing, and undergrad-level expertise in algorithms and discrete mathematics and the ability to program.

### **C S 393R. Autonomous Robots.**

Explore an advanced overview of autonomous mobile robotics, including control, perception, and planning. Examine algorithms and data structures related to these, cover widely adopted and state of the art techniques. Implement and extend such algorithms on real robots. Three lecture hours a week for one semester. [Computer Science 393R](#) and [395T](#) (Topic: Autonomous Robots) may not both be counted. Prerequisite: Graduate standing.

### **C S 394C. Algorithms for Computational Biology.**

Algorithm design in computational molecular biology, with a focus on multiple sequence alignment and phylogeny (evolutionary history) reconstruction. Topics include the design and analysis of algorithms under probabilistic models of evolution, heuristics, and exact solutions for NP-hard optimization problems. Three lecture hours a week for one semester. [Computer Science 394C](#) and [395T](#) (Topic: Algorithms for Computational Biology) may not both be counted. Prerequisite: Graduate standing.

### **C S 394D. Deep Learning.**

Explore the basic building blocks and intuitions behind designing, training, tuning, and monitoring of deep networks. Examine both the theory of deep learning, as well as hands-on implementation sessions in pytorch. Explore a series of application areas of deep networks in: computer vision, sequence modeling in natural language processing, deep reinforcement learning, generative modeling, and adversarial learning. Three lecture hours a week for one semester. Only one of the following may be counted: [Artificial Intelligence 394D](#), [Computer Science 394D](#), [Data Science 394D](#), [395T](#) (Topic: Deep Learning). Prerequisite: Graduate standing, and experience in artificial intelligence and machine learning.

### **C S 394F. Knowledge Representation and Reasoning.**

Surveys the research and practice of building knowledge systems, including knowledge representation, automated reasoning, knowledge acquisition, and explanation generation. Three lecture hours a week for one semester. Prerequisite: Graduate standing; and [Computer Science 381K](#) or the equivalent or consent of instructor.

### **C S 394N. Neural Networks.**

Same as [Neuroscience 394N](#). Biological information processing; architectures and algorithms for supervised learning, self-organization, reinforcement learning, and neuro-evolution; theoretical analysis; hardware implementations and simulators; applications in engineering, artificial intelligence, and cognitive science. Three lecture hours a week for one semester. Only one of the following may be counted: [Computer Science 394N](#), [Neuroscience 394N](#), [394P](#) (Topic: Neural Networks). Prerequisite: Graduate standing.

### **C S 394P. Automatic Programming.**

Automatic generation of computer programs from high-level specifications. Program analysis, optimization, and transformation; partial evaluation; object-oriented programming; transformation of formal specifications; specialization of generic procedures; views. Three lecture hours a week for one semester. Prerequisite: Graduate standing. [Computer Science 375](#) and [381K](#) are recommended.

### **C S 394R. Reinforcement Learning: Theory and Practice.**

Introduces the theory and practice of modern reinforcement learning, with emphasis on temporal difference learning algorithms. Three lecture hours a week for one semester. Only one of the following may be counted: [Artificial Intelligence 394R](#), [Computer Science 394R](#), [Data Science 394R](#), [395T](#) (topic: Reinforcement Learning) Prerequisite: Graduate standing.

### **C S 195, 295, 395. Conference Course.**

For each semester hour of credit earned, one lecture hour a week for one semester. Offered on the credit/no credit basis only. May be repeated for credit when the topics vary. Offered on the credit/no credit basis only. Prerequisite: Graduate standing and consent of instructor.

### **C S 195L, 295L, 395L. Conference Course.**

For each semester hour of credit earned, one lecture hour a week for one semester. May be repeated for credit when the topics vary. Prerequisite: Graduate standing and consent of instructor.

### **C S 195T, 395T. Topics in Computer Science.**

From eight to fifteen topics are offered each semester. One or three lecture hours a week for one semester. May be repeated for credit when the topics vary. Prerequisite: Graduate standing; additional prerequisites vary with the topic.

#### **Topic 1: Parallel Computations.**

**Topic 9: Introduction to Ethical AI and Robotics.** Introduction to ethical issues and challenges that could and do arise when creating AI-based systems. Explore case studies anchored by readings illustrating the current and future possible uses of robots as well as the broad themes of ethical and responsible AI. Address how things have gone wrong in the past and open debates regarding how best to design, deploy, and regulate AI systems. Additional prerequisite: Graduate standing or consent of instructor.

### **C S 396. Research Practice and Experience.**

Open only to those in their first two years as graduate students in computer science. Designed to provide an early research experience for new doctoral students in computer science. Students conduct an independent research project and present the results.

Individual instruction. May not be counted toward a master's degree in computer science. Offered on the credit/no credit basis only. Prerequisite: Graduate standing.

**C S 396M. Advanced Networking Protocols.**

Topics include routing, multiple access, internetworking, security, performance models, and verification methods. Three lecture hours a week for one semester. Prerequisite: Graduate standing.

**C S 698. Thesis.**

The equivalent of three lecture hours a week for two semesters. Offered on the credit/no credit basis only. Prerequisite: For 698A, graduate standing in computer science and consent of the graduate adviser; for 698B, [Computer Science 698A](#).

**C S 398T. Supervised Teaching in Computer Science.**

Supervised teaching experience, and seminar focused on curriculum construction and teaching methods. Three lecture hours a week for one semester. Offered on the credit/no credit basis only. Prerequisite: Graduate standing and appointment as a teaching assistant.

**C S 399W, 699W, 999W. Dissertation.**

May be repeated for credit. Offered on the credit/no credit basis only. Prerequisite: Admission to candidacy for the doctoral degree.

**Professional Courses**